PSU McNair Scholars Online Journal

Volume 1
Issue 1 *Transformative Possibilities: Transcending Interlocking Boundaries*

Article 3

2004

Automated Test Pattern Generation for Quantum Circuits

Jacob D. Biamonte
Portland State University

Recommended Citation

Biamonte, Jacob D. (2004) "Automated Test Pattern Generation for Quantum Circuits," *PSU McNair Scholars Online Journal*: Vol. 1: Iss. 1, Article 3.
10.15760/mcnair.2005.38

Automated Test Pattern Generation for Quantum Circuits

by Jacob D. Biamonte [1,2]
Faculty Mentor : Marek A. Perkowski [1,3]

[1] Department of Electrical and Computer Engineering, Portland State University
[2] Department of Physics, Portland State University
[3] Portland, OR 97207-0751, USA;
Department of Electronics and Computer Science, Korean Advanced Institute of Science and Technology, Deajeon 307-701, South Korea.

ABSTRACT

This work extends a general method used to test classical circuits to quantum circuits. Gate internal errors are address using a discrete fault model. Fault models to represent unwanted nearest neighbor entanglement as well as unwanted qubit rotation are presented. When witnessed, the faults we model are probabilistic, but there is a set of tests with the highest probability of detecting a discrete repetitive fault. A method of probabilistic set covering to identify the minimal set of tests is introduced. A large part of our work consisted of writing a software package that allows us to compare various fault models and test strategies for quantum networks.

Keywords: Automated quantum test set generation, quantum state tomography, quantum error modeling.

1. INTRODUCTION

Quantum information theory approaches computation by thinking of all physical phenomena as computations, with each measurable outcome of a computation as a system observable[1]. The past quarter century fostered an electronics revolution—with the most recent strides enabling the control of bi-state particles that can be used to represent bits of information[2]. The laws of quantum mechanics predict computational devices leading to speed increases over their classical counterparts[3]. Quantum system reliability is a very real problem[4]. Likewise, system verification has recently received quite a bit of attention[5]. Just as the EDA (Electronic Design and Automation) community began to dominate the testing of classical circuits around the 1960's[6], this paper begins to extend these classical test methods to quantum circuits. The reason it has taken so long to adapt classical testing methods to quantum circuits is because of the difference from classical test.

Currently experimental physicists have only begun to experience a need to research optimized testing methods due to the small qubit (quantum bit) count of current quantum circuits[7]. Much like in the early days of classical logic, experimentalists test using a process known as state tomography[8]—essentially brute force—where the number of tests is supper exponential in the number of qubits. In this work we consider quantum circuits that implement binary gates and oracles—we concern ourselves with the logical testing of quantum circuits. This means that we will inspect the logical data processed by the quantum network and compare this data with expected values allowing us to make a judgment on a circuit's logical functionality—after all we measure only binary information from a qubit even though its quantum modes of operation reside in the vastness of Hilbert Space.

The relatively slow rate of progress realizing quantum circuits causes some to consider our idea of fast testing a bit premature. On the other hand, we can say that in NMR it takes months to fine tune the sequence of pulses necessary to implement a simple universal gate to function properly[9], and much of this time is spent testing. Therefore, improvements over current testing methods would foster development towards a quantum information processing system.

Classically test set generation relies on a fault model, this means that you limit the errors to those that are most likely. In this work, we consider internal quantum gate faults such as unwanted qubit rotations and unwanted entanglement. In practice, the choice of the fault model will be determined by a particular quantum

J.B. is the author with whom electronic correspondence shall be addressed: biamonte@ieee.org

circuit technology, but our model is general enough to cover a wide range of fault types. Often faults in quantum circuits are constant in space and time[10], this could be thought of as a presence of an unwanted field. However to simplify our approach in this introductory work, we limit our discussion to discrete faults occurring in the timed operations forming a quantum circuit. Faults other than local decoherence have been studied recently. Kak[11] addressed the problems in initialization of quantum computers, Shenvi et al addressed the Grover search under the impact of noise[12], and Bettelli modelled the impact of non-ideal operations on decoherence[13].

We proceed now to Section 2 where we present the necessary background in quantum mechanics to read the paper. This is followed by Section 3 where the error models and fault table generation method is presented. In Section 3.1 our method to generate test plans based on a given quantum fault table is presented, this is followed by Section 3.2 where our set covering algorithm is formalized. Section 4 concludes the paper by a discussion of the validity of this approach based on the data generated by QuFault, our quantum test set generation software package. In this paper we often omit normalization factors and represent $|+\rangle$ as $|0\rangle + |1\rangle$ and $|-\rangle$ as $|0\rangle - |1\rangle$ for clarity.

2. BACKGROUND

In quantum computing the bi-state subatomic-particles used to store data are called qubits. Just like in classical computing, information is processed by gates, however in most quantum computing architectures a gate is an operator that acts on the state space of a system, an example of such an operator is an electromagnetic pulse in NMR[8]. This can become confusing to the electrical engineer who thinks of gates as those occupying space, with information represented by that of a signal acted on in the course of time. In quantum computing however, gates act on qubits that contain information and occupy space. A qubit takes continuous values in space with time, making information processing capabilities with qubits intriguing as one can place a single qubit in a superposition of classical states. Yet when a qubit is measured, its complex state is destroyed leaving the system in a single eigenstate, and returning an eigenvalue of an observable. The gates in quantum computing are represented as unitary operators, and the state of a quantum circuit is described as a unit vector in a Hilbert Space.

To define a qubit one must specify an orthonormal vector space to both describe the system, and give reference to measurement operations. The state of a single qubit is represented as a point on the sphere *(for pure states)* formed in a Hilbert space of 4 dimensions. We can make a measurement along our choice of the orthonormal x, y or z axis. The outcome of a measurement will be plus or minus one, the eigenvalues of an observable. The z axis is what is known as the computational basis, and is generally implied when communicating ideas about quantum computing, like when we describe the state of the following system: $|\Psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$, where $|0\rangle = (1\ 0)^T$ and $|1\rangle = (0\ 1)^T$. The weights alpha and beta combined with the Eigenstates $|0\rangle$ and $|1\rangle$ form an orthonormal basis, where $\alpha\alpha^* + \beta\beta^* = 1$. The x axis (Also known as the plus minus basis) is used sometimes in quantum computing to describe qubits. Typically the plus minus basis is denoted as $|+\rangle$ and $|-\rangle$. One can express this new basis in terms of the computational basis as, $|+\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$, and $|-\rangle = \frac{|0\rangle-|1\rangle}{\sqrt{2}}$. Even more generally we can represent a qubit with Equation 1.

$$|\Psi\rangle = cos(\theta) \cdot |0\rangle + e^{-\jmath\phi} sin(\theta) \cdot |1\rangle \tag{1}$$

If we examine Equation 1 in more detail we will notice that a qubit has what is called both amplitude and phase - where the $e^{-\jmath\phi}$ term represents the qubit phase. The amplitude corresponds to the probability of measuring the system to be found in a given eigenstate. In the computational basis we can not detect phase, and furthermore an operator acting on a qubit will rotate the phase based on the eigenvalues of the operator. However, phase can be detected from the plus minus basis, and we note that phase is imperative in quantum algorithms[8].

The operators given in Equations 9, 10 and 11, from a basis space for measurement, as well as gates that preform unitary evolution in quantum computing. When we substitute values of $\theta = 2\pi$ into Equations 9, 10 and $\phi = 2\pi$ into Equation 11 we obtain what is known as the Pauli matrices given in Equations 2, 3 and 4.

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{2}$$

$$\sigma_y = \begin{pmatrix} 0 & -\jmath \\ \jmath & 0 \end{pmatrix} \tag{3}$$

$$\sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{4}$$

Observe, that the eigenvalues of each of the Pauli matrices are plus or minus one and that each eigenvector can represent an axis of a Hilbert Space used to describe a single qubit. The size of the Hilbert space used to represent the state of any quantum system expands dimensionally via the tensor product[14]. The state space of an n qubit quantum computational system corresponds to 2^n dimensions, and a given state vector in that space can be in 2^n different states at any given time, where a classical computational system can reside in only a single state. Linear combinations of the Pauli Matrices can be used to create other useful gates, such as the Hadamard gate whose unitary matrix is given in Equation 5.

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{5}$$

We now have the building blocks needed to represent simple quantum computations. To represent a quantum computation we must initialize a qubit into a certain state before we act on that qubit to process information. For example, typically we rotate the state of a qubit such that it will be "+1" in the computational basis before computation. In other words the state of our system is described as $|\Psi\rangle = |0\rangle$ and the inner product between the positive z axis and the state of our qubit is the unit value "$+1$". After we initialize our qubit we can act on it with a Hardmard operation, such as:

$$H|\Psi\rangle = \frac{1}{\sqrt{2}} \cdot \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \cdot |0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} = |+\rangle \tag{6}$$

This operation mapped our system into what is known as a superposition of states, both possible eigenstates in time. We can now use projective measurement and project the state of our system onto the computational basis using the σ_z observable. We can calculate the expectation value of this observable as follows:

$$\langle +|\sigma_z|+\rangle = \frac{1}{\sqrt{2}} \cdot \begin{pmatrix} 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \cdot \frac{1}{\sqrt{2}} \cdot \begin{pmatrix} 1 \\ 1 \end{pmatrix} = 0 \tag{7}$$

This means that if we measure our system from the computational basis infinitely many times we will get value "+1" half of the time and value "-1" the other half of the time and the average is 0. If we instead made a measurement of the expectation value of the σ_x observable we would have obtained the following,

$$\langle + | \sigma_x | + \rangle = \frac{1}{\sqrt{2}} \cdot \begin{pmatrix} 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \cdot \frac{1}{\sqrt{2}} \cdot \begin{pmatrix} 1 \\ 1 \end{pmatrix} = +1 \tag{8}$$

The measurement outcome is what is referred to as 'sharp' in σ_x since it will always yield value +1.

This section concludes the provided background information on quantum computation. It is designed as an overview but we refer the reader to[1], as well as[14] and[8] for more depth of subject coverage.

3. MODELLING ERRORS

Traditionally the Test Generation Problem is thought of as the generation of a sequence of tests (test set), that when applied to a circuit and compared with the circuit's output, will determine that the circuit is correct or will determine that it contains one or more faults[15]. In other words, testing is the checking of functionality, and running the ideal test set amounts to sufficient system verification with the smallest possible number of tests[6].

For simplification, in this paper we assume what is called in binary testing the single fault model[6], where only one fault is allowed in the entire circuit. In classical test you first determine what output the circuit should generate under certain conditions and then find out what the circuit generate if certain errors were present, such as output a $|001\rangle$ instead of a $|101\rangle$ ten percent of the time with input vector $|000\rangle$. In this work we consider *'what if'* cases, where a model of the error is placed into a circuit and the new erroneous output is calculated and stored—for later comparison. We then use software to find the smallest set of tests that can detect these errors to a certain level of validation, referred to as ϵ. For the quantum case this table contains fractions that represent probabilities of different outcomes, as will be seen in section 3.1. To that order, we must distinguish between a probabilistic fault and a deterministic fault that is *observed* probabilistically. When we wrote this paper we coined these faults "quantum faults" in order to better distinguish them from their classical counterparts.

The Hamiltonian of the spin system that models Ising type interactions is given by, $\hat{H} = \sum_{i,\alpha} \alpha_{i\alpha}\sigma_{i\alpha} + \sum_{i,j} J_{ij}\sigma_{i\alpha}\sigma_{j\alpha}$. The spin Hamiltonian governs the ideal operation of a quantum circuit. It contains a term for individual bitwise operations, $R_{i,\alpha}(\theta) = e^{-\jmath\frac{\phi}{2}\sigma_{i\alpha}}$ and an interaction (Ising term) $J_{ij}(\phi) = e^{-\jmath\frac{\phi}{2}\sigma_{i\alpha}\sigma_{j\alpha}}$ to allow entanglement between qubits. Similar forms of this equation are relevant for any quantum computer. For our purposes, in the time dilation of pulses governed by the Hamiltonian, we assume an additional term representing a small error. The resulting impact in the presence of this error manifests its self in the form of changing the probability amplitudes of the possible outcomes, thus there is an altered chance for a particular outcome in measurement. A fault present in the state vector representing the system takes one of two forms, the first being the observation or (measurement/detection), and the second being the lack of observation.

Another difficulty detecting errors in quantum circuits is due to an error in the phase of the qubit. In the computational basis we can not detect phase because the eigenvectors of phase errors are the eigenvectors of the computational basis. The eigenvalues of a fault impacting the phase of a qubit will rotate the phase based on the eigenvalues of the fault. The interesting fault model of gate removal originates from Hayes et al who applied it to remove entire gates such as Toffoli gates in reversible circuits[16]. We believe however that this model is more adequate to single pulses and not gates composed of many pulses. Thus, a permutation circuit can become non-permutating as the result of a fault. The fault model presented here assumes that the machine is firing pulses, and that these could contain errors themselves, or another unwanted interaction changes the state of a qubit. The impact these faults have on the state of a qubit must be below a bound that is acceptable if the machine is going to function as expected. For example, a short or long pulse or a refocusing error in NMR can result in unwanted qubit rotations. This fault can be modelled in the simplest terms with a single qubit rotation operator about the x, y and z axis. If we consider the circuit shown in Figure 1 as a sequence of stages with each

stage defined as a gate that does not commute with both nearest neighbors, then the circuit shown in Figure 1 has 5 stages. We can assume certain faults between any of the stages or internal to the stages themselves in the circuit, in Figure 1 the locations of possible faults are represented by placing an " × " on the wire.

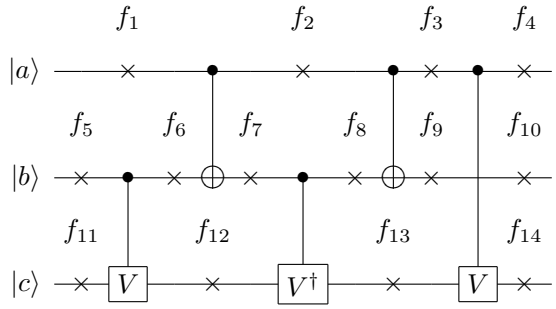

Figure 1. *Toffoli gate made with smaller gates, the next level below this is machine dependent pulses. For a better explanation of this gate we refer the reader to[8].*

We model the faults impacting the gate shown in Figure 1 by replacing each " × " with a fault model represented by a matrix. The circuits transfer matrix is then recalculated and stored for later comparison. We calculate the probability outcomes for all inputs of a good circuit based on the unitary transfer matrix of a circuit, we then store this in the fault table. For each fault model in our library and for each " × " in the circuit, we insert the matrix representing the fault in place of the " × " and recalculate the transfer matrix of the circuit. We then store the difference in the probabilities of the correct circuit response for a given input and the altered circuit response under a given fault. We find the input test set that reveals these faults with the highest probability using the probabilistic set covering method from Section 3.1. This method is general enough to work with many fault models. Quantum error correcting codes are typically designed to correct from insertions of the gates given in Equations 9, 10 and 11. This fault model has been used by[12] as well as several other papers.

$$R_x(\theta) = \begin{pmatrix} cos(\frac{\theta}{2}) & -\jmath \cdot sin(\frac{\theta}{2}) \\ -\jmath \cdot sin(\frac{\theta}{2}) & cos(\frac{\theta}{2}) \end{pmatrix} \tag{9}$$

$$R_y(\theta) = \begin{pmatrix} cos(\frac{\theta}{2}) & -sin(\frac{\theta}{2}) \\ sin(\frac{\theta}{2}) & cos(\frac{\theta}{2}) \end{pmatrix} \tag{10}$$

$$R_z(\phi) = \begin{pmatrix} e^{-\jmath\frac{\phi}{2}} & 0 \\ 0 & e^{\jmath\frac{\phi}{2}} \end{pmatrix} \tag{11}$$

We can also use the tensor product to expand any number of these single bit operators to show a simple case where a fault *spreads out* and impacts other bits in the circuit. The table shown in Figure 2 was generated using the fault models given in Equations 9, 10 and 11. Repeating rows and columns were grouped for clarity, and $|+\rangle = |0\rangle + |1\rangle$ and $|-\rangle = |0\rangle - |1\rangle$. Each entry in the table corresponds to the probability of detecting a given fault represented by Column label f_n. Equations 9, 10, and 11 correspond to single bit rotations. The angle of this rotation is given as θ in Equations 9 and 10 and ϕ in Equation 11. So if we we can represent the margin of error as some ε, we observe that $\frac{|\theta|}{4\pi} = \frac{\varepsilon}{100}$, and solving for θ gives us, $\theta = \frac{\pm\varepsilon\pi}{25}$. If ε is substituted into Equations 9, 10 or 11 than it can be thought of as a margin or percentage of error. For the Toffoli gate shown in Figure 1, we set ε to be 25, and then we use our software package QuFault to generate the fault table from Figure 2.

Input Test	f_a	f_b	f_c	f_d	f_e	f_f	f_g	f_h	f_i	f_j	f_k
$\|000\rangle, \|001\rangle$	0.2	0.2	0.2	0.2	0	0.2	0.2	0.2	0	0	0
$\|010\rangle, \|110\rangle$	0.2	0.2	0.2	0.2	0	0.2	0	0	0	0.2	0.2
$\|100\rangle, \|101\rangle$	0.2	0.2	0.2	0.2	0	0.2	0.2	0	0	0	0.2
$\|110\rangle, \|111\rangle$	0.2	0.2	0.2	0.2	0	0.2	0	0.2	0	0.2	0
$\|++\rangle, \|+-\rangle, \|-+\rangle, \|--\rangle$	0	0	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2
$\|++-\rangle, \|+--\rangle, \|-+-\rangle, \|---\rangle$	0	0.8	0.8	0	0	0.2	0.2	0.2	0.2	0.2	0.2

Figure 2. *Fault table generate using the fault models given in Equations 9, 10 and 11 for the Toffoli gate from Figure 1. Each entry in the fault table was created with the following equation:* $E_{t_{t f_n}} = |(\sum_{ii} GC\,|\Psi_i\rangle\langle\Psi_i|\,GC^{\dagger} - BC_n\,|\Psi_i\rangle\langle\Psi_i|\,BC_n^{\dagger})|$, *GC is the operator corresponding to a faultless circuit, BC is the operator corresponding the the impact of a given fault. Each entry in the fault table is the magnitude of the difference between GC and BC.*

To detect phase errors we need to use inputs of type: $|+\rangle = |0\rangle + |1\rangle$ and $|-\rangle = |0\rangle - |1\rangle$. One of the more interesting fault models that we used in this study represents unwanted amounts of entanglement. This fault model given in Figure 3 can model both the addition of unwanted entanglement and the removal of wanted entanglement. The amount of entanglement (the tangle τ) can be determined and adjusted using the Haar measure. the Haar measure is a well known metric used to calculate entanglement discussed in any advanced text on quantum mechanics. In Figure 4 we present a fault table generated by QuFault using unwanted entanglement as the fault model, in this case the Haar measure was set to $\tau = 1$, or maximum tangle.

$$\begin{pmatrix} \jmath \cdot e^{(\jmath\phi)} cos(\frac{\theta}{2}) & 0 & -\jmath \cdot e^{(\jmath\phi)} sin(\frac{\theta}{2}) & 0 \\ 0 & \jmath \cdot e^{(\jmath\phi)} cos(\frac{\theta}{2}) & 0 & -\jmath \cdot e^{(\jmath\phi)} sin(\frac{\theta}{2}) \\ \jmath \cdot e^{(-\jmath\phi)} sin(\frac{\theta}{2}) cos(\theta) & e^{(-\jmath\phi)} sin(\frac{\theta}{2}) \cdot sin(\frac{\theta}{2}) & \jmath \cdot e^{(\jmath\phi)} cos(\frac{\theta}{2}) \cdot cos(\theta) & e^{(\jmath\phi)} cos(\frac{\theta}{2}) sin(\theta) \\ e^{(-\jmath\phi)} sin(\frac{\theta}{2}) \cdot sin(\frac{\theta}{2}) & \jmath \cdot e^{(-\jmath\phi)} sin(\frac{\theta}{2}) cos(\theta) & e^{(\jmath\phi)} cos(\frac{\theta}{2}) sin(\theta) & \jmath \cdot e^{(\jmath\phi)} cos(\theta) \cdot cos(\frac{\theta}{2}) \end{pmatrix} \tag{12}$$

Figure 3. *The fault model used to represent the unwanted interaction of two qubits. We can vary the amount of entanglement over the range of the Haar Measure, 0 tangle means no entanglement and 1 means maximum tangle.*

Input Test	f_1	f_2	f_3	f_4	f_5	f_6
$\|000\rangle, \|001\rangle$	0.5	0.5	0.5	0.5	0.5	0.5
$\|010\rangle, \|110\rangle$	0.5	0.5	1	1	1	1
$\|100\rangle, \|101\rangle$	1	1	0.5	0.5	1	1
$\|110\rangle, \|111\rangle$	1	1	1	1	0.5	0.5

Figure 4. *A fault table created with the Toffoli gate from Figure 1 contains internal errors resulting in unwanted entanglement. A representation of the fault model is given in Figure 3. Here the values of* $\theta = \pi/2$ *and* $\phi = \pi$ *making Figure 3 become the well known operator used to create Bell states.*

3.1. Quantized Set covering

Because of the fractions that occur in quantum fault Tables 2 and 4, it is clear that traditional set covering will not work. In order to detect the greatest number of faults with each test, we must introduce a new kind of set covering.

We represent by a_n an entry in the quantum fault table, it represents the probability that given row (test) covers given column (fault), $0 \leq a_n \leq 1$. It is clear that the traditional set covering problem of a fault table is a special case where $\forall a_n, a_n \in \{0,1\}^n$. The quantum set covering problem is now formulated differing from the classical case with the addition of positive fractional entries, arising from the nature of quantum measurement. This modification in problem formulation makes the concept of full coverage in general not achievable for testing quantum circuits, unless we specify a bound ϵ used as an accuracy cut off. Given the Table from Figure 5, for every selected set of rows one can calculate the probability of detecting fault f_a. Assume that rows T_a, T_{a+1} and T_{a+2} have been selected as a (quasi)-solution to the probabilistic set covering problem being a sub-problem of quantum set covering. Then the probability of detecting fault f_a is $P(f_a) = P(a_1) + (1 - P(a_1)) \cdot P(b_1) + (1 - (1 - P(a_1)) \cdot (1 - P(b_1)) \cdot P(c_1)$. If the probability of detecting fault f_a is above a certain "accuracy level" for each column, then the set of tests is a solution. If not, one can select other tests or repeat some tests T_j k times to increase the probability until we reach the desired assurance. Assume that in column f_r only test T_1 has entry higher than 0, and that this value is 3/4 . Observe that by repeating this test three times we get the probability $P(f_r)$ of detecting fault f_r defined by $P(f_r) = \frac{3}{4} + \frac{1}{4} \cdot \frac{3}{4} + \frac{1}{4} \cdot \frac{3}{4} \cdot \frac{3}{4} = \frac{3}{4} + \frac{15}{64}$. We increased thus the fault detection probability by 15/64 by repeating the test three times. The techniques presented above are used in quantum set covering. In many cases the quantum fault table has a high percent of columns with "1's" thus can be highly reduced by using classical set covering approaches based on dominance and equivalence[6].

$\ddots$	f_a	f_{a+1}	f_{a+2}	$\dots$	f_n
$\hat{T}_a$	$P(a_1)$	$P(a_2)$	$P(a_3)$		$P(a_n)$
$\hat{T}_{a+1}$	$P(b_1)$	$P(b_2)$	$P(b_3)$	$\dots$	$P(b_n)$
$\hat{T}_{a+2}$	$P(c_1)$	$P(c_2)$	$P(c_3)$		$P(c_n)$
$\vdots$		$\vdots$		$\ddots$	$\vdots$
$\hat{T}_n$	$P(n_1)$	$P(n_2)$	$P(n_3)$		$P(n_n)$

Figure 5. *Arbitrary Fault Table with Columns as faults and Rows as tests and entries representing the probability of a given test detecting a given fault.*

3.2. Quantum set covering formulation

The Quantum Set Covering Problem is formulated as follows. Given is a covering table in which rows correspond to tests and columns to faults. The value p in the entry on the intersection of the row R and column C means that test R detects fault C with probability p. Value 1 in the entry on the intersection of row R and column C means probability 1 or that R is a deterministic test for C.

Rule 1. The entries of a quantum fault table are governed by a constrained covering problem with fractional entries. The entries relate to each other with the inclusion/exclusion principle and in discrete form are governed by the following Equation: $c(n) = \sum_{n=1}^{N} (1 - c(n-1)) \cdot a_n$.

Definition 1. Two tests (rows) R_i and R_j of a Quantum Fault Table are equivalent if they are identical vectors of numbers.

Definition 2. In a Quantum Fault Table test T_1 dominates test T_2 if **every** entry of test T_2 is a smaller number than the corresponding entry in the same column of test T_1.

Dominated rows can be removed. In a group of equivalent rows, all but one can be removed. The algorithm below takes into account the "row equivalence" and "row domination" defined as Definition 1 and Definition 2 above, differing

from standard set covering algorithms. The classical set covering algorithm is a special case of quantum set covering, thus the Definitions 1 and 2 still hold for standard set covering, but not vice versa.

Definition 3. Assurance of column C_j with respect to set RR of rows is the sum of products of probabilities $c_{ij}(n)$ taken for all rows R_j from RR (as illustrated in 3.1 for $P(f_a)$ for rows T_a, T_{a+1} and T_{a+2}.) The "achieved assurance" of a column is the assurance for the set R of rows selected to the solution. The total assurance is the sum of achieved column assurances calculated for all columns and for the set of selected rows from the solution (including the rows selected at stages 1-3 of the algorithm below). Our heuristic greedy algorithm to solve the Quantum Set Covering Problem applied to the original Fault Table (Table S1) is as follows.

Quantum Set Covering Algorithm: Start from a given Quantum Fault Table S1.

1: Remove all dominated rows and select one row in each group of equivalent rows. Create Table S2.

2: Find a solution SOL1 (using any known set covering algorithm) to the subset of the table composed of all columns that have at least one "1" in them and their respective rows. Remove from Table S2 the rows from SOL1 and columns that are covered in SOL1. Remove dominated rows and select one row in each group of equivalent rows. Create Table S3.

3: Assume given accuracy acc (0,1). It can be different for each column C_m, denoted by column accuracy *accm*.

4: Create a vector of 0's the column length equal to S3, denoted this vector COL.

Process of Table Reduction:

I. Define x as the number of entries in each row in S2.

II. Calculate the inner product between COL and S2, $\sum_x \sqrt{COL_x \cdot S2_x}$.

II. Select the row in S2 as a test, where the inner product between COL and S2 is maximum.

III. Recalculate the entries of COL using Rule 1.

IV. For each of the x entries in COL: if COL_x remove entry x from COL and S3.

V. Repeat Process while $x > 0$.

VI. Determine if the classical solution was dominated in the other selected tests.

The condition of satisfying accuracy acc_j for each column is a good termination condition because it is satisfiable (from problem formulation there always exist some subset of rows that satisfies this condition). In addition, the search uses the cost function which is the maximal total assurance, based on the amount of information gained by a potential test. The total assurance is calculated for the set of selected rows from the solution (including the rows selected at stages 1-3). Whenever a new value of the total assurance is found, if it is larger than the previously stored value, the corresponding solution is retained together with its total assurance. This way, when the search is completed, the last solution has the maximum value of the total assurance among all solutions that satisfy the termination condition of all "column accuracies" acc_j . The final solution is the union of SOL1 and SOL2. In[19] the notion of probabilistic set-covering is introduced as the generation of a random binary vector and the covering constraint has to be satisfied with some prescribed probability. Although there is certain similarity, our "quantum set covering problem" is quite different. Some ideas of[19] can be however used to create other algorithms for our problem.

4. CONCLUSIONS, COMPARISONS AND FUTURE WORK

We addressed the problem of generating test patterns for quantum circuits. We presented an algorithm to minimize the number of tests for quantum circuits based on an extension of the classical set covering algorithm. The presented method represents a possible solution to the problem of quantum test set generation. The weakness of the method lies in validation of the fault model. Although repetitive faults seem very possible in a quantum information processing system, faults may well be constant in time and space. More generalized fault models specific to quantum computing technology must be addressed by the Quantum EDA community, QEDA. This will be the direction of the authors future research. A somewhat obvious fact we noted in this work is that short fat channels are easier to test than long skinny ones. We noticed also that when taking a cross section of 4 bits and generating random stages, the number of tests to verify the network grows as a log with the number of stages.

Circuit	$\epsilon = 50$	$\epsilon = 25$	$\epsilon = 15$
Peres	3	9	22
Toffoli	3	9	21
Miller	4	10	23
Fredkin	3	9	21

Figure 6. Comparison on the number of tests needed for some common circuits.

5. ACKNOWLEDGMENTS

Portland State University provided funding, resources, space and support for this project. Thanks are also due to George Fredrick Viamontes and Prof. Igor L. Markov both from the University of Michigan, for providing us with the quantum simulation tool QuIDDPro and for comments and encouragement on the draft version of this paper. The package Qcircuit.tex produced all the circuits in this text, details can be found at, http://xxx.lanl.gov/abs/quant-ph/0406003. J.D.B and M.A.P. received support by Ronald E. McNair Post baccalaureate Achievement Program of Portland State University, and the Korean Institute of Science and Technology, during parts of this project. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing official policies of endorsements, either expressed or implied, of the Funding agencies.

REFERENCES

1. D. Deutsch, *Video Lectures on Quantum Computation*, (2004-05), `www.quiprocone.org`.
2. J. Kim, J-S. Lee, S. Lee and C. Cheong, *Implementation of the refined Deutsch-Jozsa algorithm on a 3-bit NMR quantum computer*, Phys. Rev. A 62, 022312 (2000), `quant-ph/9910015`.
3. P. W. Shor, *Introduction to Quantum Algorithms*, (1999), `quant-ph/0005003`.
4. J. Preskill, *Reliable quantum computers*, Proc. R. Soc. London A, 454(1969):385-410, 1998.
5. J. Biamonte, M. Perkowski, *Testing a Quantum Computer*, Proceedings of 2004 KIAS-KIST, WQIS, Soule Korea, August, 2004; J. Biamonte, J. Buell, M. Perkowski, *Quantum Fault Localization*, INQ Research Conference, Portland Oregon, May 19, 2004; M. Perkowski, J. Biamonte, M. Lukac, *Test Generation and Fault Localization for Quantum Circuits*, to appear, ISMVL 2005; S. Aligala, S. Ratakonda, K. Narayan, K. Nagarajan, M. Lukac, J. Biamonte and M. Perkowski, *Deterministic and Probabilistic Test Generation for Binary and Ternary Quantum Circuits*, Proceedings of ULSI 2004, Toronto, Canada, May 2004; J. Biamonte, M. Jeoung, M. A. Perkowski, *Extending Classical Test to Quantum-Theory and Experiment*, to appear Proc. Of SPIE International Conference on Fluctuations and Noise,, SPIE Paper Number 5842-43, May 23-26 2005, Austin Texas USA.
6. C. Landrault, Translated by M. A. Perkowski, *Test and Design For Test*, `ee.pdx.edu/`.
7. D. Vion, A. Aassime, A. Cottet, P. Joyez, H. Pothier, C. Urbina, D. Esteve, M.H. Devoret, *Manipulating the Quantum State of an Electrical Circuit*, (2003), `cond-mat/0304232`.
8. M. Nielsen and I. Chuang, *Quantum Computation and Quantum Information*, Cambridge Univ. Press, 2000.
9. J. Kim, J-S. Lee and S. Lee, *Implementing unitary operators in quantum computation*, Phys. Rev. A 61 (2000) 032312, `quant-ph/9908052`.
10. D. Aharonov, M. Ben-Or, *Fault-Tolerant Quantum Computation With Constant Error Rate*, `quant-ph/9906129`.
11. S. Kak, *The Initialization Problem in Quantum Computing*, Foundations of Physics, vol 29, pp. 267-279, 1999.
12. N. Shenvi, K. R. Brown, K. B. Whaley, *Effects of Noisy Oracle on Search Algorithm Complexity*, `quant-ph/0304138`.
13. S. Bettelli, *Quantitative model for the effective decoherence of a quantum computer with imperfect unitary operations*, Physical Review A 69,042310, 14 pages, 2004.
14. M. Oskin, *Quantum Computing Lecture Notes*, Notes to CSE590mo, University of Washington, (2004), `cs.washington.edu`.
15. K. N. Patel, J. P. Hayes and I. L. Markov, *Fault Testing for Reversible Circuits*, IEEE Trans. on CAD, 23(8), pp. 1220-1230, August 2004, `quant-ph/0404003`.
16. J.P. Hayes, I. Polian, B. Becker, *Testing for Missing-Gate Faults in Reversible Circuits*, Proc. Asian Test Symposium, Taiwain, November 2004.
17. G. F. Viamontes, I. L. Markov and J. P. Hayes, *Improving Gate-Level Simulation of Quantum Circuits*, Quantum Information Processing, vol. 2(5), October 2003, pp.347-380. `quant-ph/0309060`.
18. S. Lee, J-S. Lee, T. Kim, J. Biamonte and M. Perkowski, *The Cost of Quantum Gate Primitives*, in review, 2004.
19. P. Beraldi, A. Ruszczynski, *The Probabilistic Set Covering Problem*, Operations Research 2002 INFORMS, Vol.50, No.6, November-December 2002, pp. 956-967.

20. J. Biamonte and M. Perkowski, *Testing a Quantum Computer*, Proceedings of, KAIS, Workshop on Quantum Information Sicence, Seoul Korea, August 29th - 31st, 2004.
21. K. M. Obenland, and A. M. Despan, *Impact of Errors on a Quantum Computer Architecture* Technical Report, Information Sciences Institute, University of Sothern California, Oct 1st, 1996.
22. G. F. Viamontes, I. L. Markov, J. P. Hayes, *Graph-based simulation of quantum computation in the density matrix representation*, Quantum Information and Computation, 5 (2), pp. 113-130, (2005), `quant-ph/0403114`.